EXHIBIT L

FILED UNDER SEAL

CLEMENT SETH ROBERTS (STATE BAR NO. 209203)
croberts@orrick.com
BAS DE BLANK (STATE BAR NO. 191487)
basdeblank@orrick.com
ALYSSA CARIDIS (STATE BAR NO. 260103)
acaridis@orrick.com
EVAN D. BREWER (STATE BAR NO. 304411)
ebrewer@orrick.com
ORRICK, HERRINGTON & SUTCLIFFE LLP
The Orrick Building
405 Howard Street
San Francisco, CA 94105-2669
Telephone: +1 415 773 5700
Facsimile: +1 415 773 5759

SEAN M. SULLIVAN (*pro hac vice*)
sullivan@ls3ip.com
COLE RICHTER (*pro hac vice*)
richter@ls3ip.com
LEE SULLIVAN SHEA & SMITH LLP
656 W Randolph St., Floor 5W
Chicago, IL 60661
Telephone: +1 312 754 0002
Facsimile: +1 312 754 0003

Attorneys for Defendant Sonos, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

GOOGLE LLC, Plaintiff and Counter-defendant, v. SONOS, INC., Defendant and Counter-claimant.	Case No. 3:20-cv-06754-WHA Related to Case No. 3:21-cv-07559-WHA **FIRST SUPPLEMENTAL REPLY EXPERT REPORT OF DR. KEVIN C. ALMEROTH**

65. The table below summarizes my observations of the three above-described scenarios for creating a new speaker group of Accused Google Players installed with newly-released firmware version 1.56.324896 and/or firmware version 1.63. As set forth in the table below, the first column describes the group creation scenario in terms of the operating modes and playback states of Accused Google Players before being added to a new speaker group; the second column describes the operating modes and playback states of the Accused Google Players after being added to the new speaker group; and the third column notes whether Dr. Schonfeld addressed the group creation scenario in his Rebuttal Report.

66.

Creating a New Speaker Group		
Group Creation Scenario	**Group Creation Result**	**Schonfeld Rebuttal Report?**
Scenario #1		
First Accused Google Player operating in standalone mode and *not* engaging in active playback Second Accused Google Player operating in standalone mode and individually engaging in active playback	First Accused Google Player continues operating in standalone mode and continues to *not* engage in active playback Second Accused Google Player continues operating in standalone mode but stops engaging in active playback	Yes - *see* Schonfeld Rebuttal Report at ¶¶ 55, 57[6]
Scenario #2		
First Accused Google Player operating in standalone mode and individually engaging in active playback Second Accused Google Player operating in standalone mode and	First Accused Google Player continues operating in standalone mode but stops engaging in active playback Second Accused Google Player continues operating in standalone	Yes - *see* Schonfeld Rebuttal Report at ¶ 56

[6] Paragraph 57 of Dr. Schonfeld's Rebuttal Report describes a scenario where one Accused Google Player is not engaging in active playback and two Accused Google Players are engaging in active playback at the time that they are all added to a new speaker group. However, based on the testing I oversaw and directed, regardless of the number of Accused Google Players that are not engaging in active playback or are individually engaging in active playback at the time that they are added to a new speaker group, each Accused Google Player not engaging in active playback continues to operate in standalone mode and continues to not engage in active playback, and each Accused Google Player that is individually engaging in active playback continues to operate in standalone mode but stops engaging in active playback.

individually engaging in active playback	mode but stops engaging in active playback	
Scenario #3		
First Accused Google Player operating in standalone mode and *not* engaging in active playback Second Accused Google Player operating in standalone mode and *not* engaging in active playback	First Accused Google Player continues operating in standalone mode and continues to *not* engage in active playback Second Accused Google Player continues operating in standalone mode and continues to *not* engage in active playback	No

67. As shown and described above and in **Appendix A**, an Accused Google Player installed with newly-released firmware version 1.56.324896 and/or firmware version 1.63 that is operating in standalone mode is capable of being added to a new speaker group regardless of whether or not the Accused Google Player is engaging in active playback. This functionality provided by newly-released firmware version 1.56.324896 has not changed from the prior firmware versions that I analyzed in my 11/30/2022 Opening Report. *See, e.g.*, 11/30/2022 Almeroth Opening Report at ¶¶ 199-226.

68. As also shown and described above and in **Appendix A**, when an Accused Google Player installed with newly-released firmware version 1.56.324896 and/or firmware version 1.63 is operating in standalone mode and not engaging in active playback at the time the Accused Google Player is added to a new speaker group, the observable behavior of the Accused Google Player will not change – it will continue to operate in standalone mode and will continue to not engage in active playback after the new speaker group is created and saved. In this respect, the behavior of an Accused Google Player that I observed in these scenarios is no different from how an Accused Google Player behaved when installed with the prior firmware versions that I analyzed in my 11/30/2022 Opening Report. *See, e.g.*, 11/30/2022 Almeroth Opening Report at ¶¶ 199-226; 1/25/2023 K. MacKay Rough Dep. Tr. at 50:25-51:24 (Google's corporate designee confirming that in a scenario where an Accused Google Player being added to a new speaker group is not running a current receiver app, there will be no change to the "operational behavior" of the

Google Player for receiving and handling a "join_group" message for a new speaker group that is sent by an Accused Google Controller after the Accused Google Player has been added to the new speaker group:

[redacted]

See 11/30/2022 Almeroth Opening Report at ¶ 459.

89. While I did not expressly memorialize it in my 11/30/2022 Opening Report, the foregoing source code path for prior firmware version 1.50 also continues on, with MultizoneManager::OnMultizoneGroupConfigChanged() calling a function named MultizoneManager::RefreshDeviceGroups() and MultizoneManager::RefreshDeviceGroups() in turn calling a function named MultizoneManager::AddGroup(). *See* multizone_setup.cc in /chromecast_internal-1.50/multizone/manager/. *See* 1/25/2023 K. MacKay Rough Dep. Tr. at 9:14-10:4, 23:5-8 (Google's corporate designee confirming that in previous versions of the source code, there was a call to the RefreshDeviceGroups() function that in turn called the AddGroup() function).

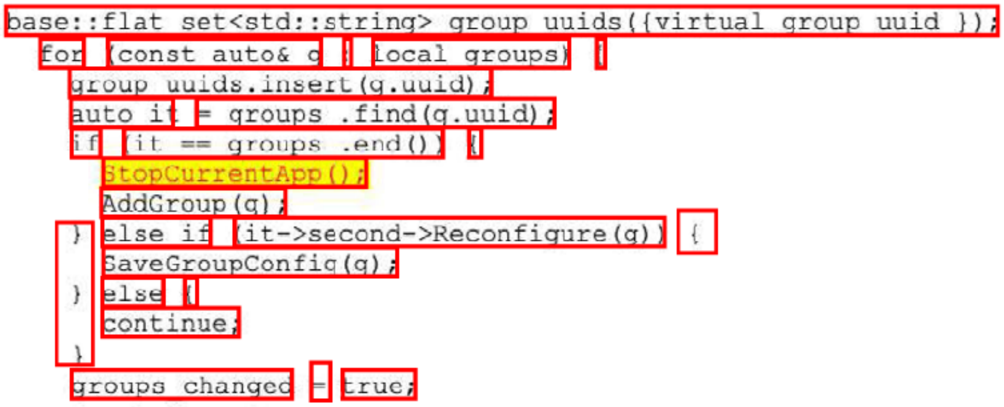
90. As noted in my 11/30/2022 Opening Report, my understanding is that this source code is representative of the source code for the firmware versions that had been released by Google from the time that the infringing timeframe began on November 5, 2019 until the November 30, 2022 date of my Opening Report.

91. Further, there is no dispute that the foregoing source code path, which is representative of the functionality carried out by an Accused Google Player installed with prior firmware version when receiving a "join_group" message indicating that the Accused Google Player has been added to a new speaker group, does not cause an Accused Google Player operating in a standalone mode to transition into a grouped mode in which it operates in accordance with the new speaker group. Rather, the foregoing source code path causes an Accused Google Player to update its internal records to memorialize that it has been added to the new speaker group by

storing information about the new speaker group while the new speaker group remains in an unlaunched state and the Accused Google Player remains in standalone mode. This is confirmed by the Court's summary judgment ruling, which found that Accused Google Players installed with firmware versions available as of July 21, 2022 – which were based on source code that included the foregoing source code path – satisfied the limitation of Asserted Claim 1 of the '885 Patent requiring the Accused Google Players to "continu[e] to operate in the standalone mode" after receiving the "join_group" messages accused of meeting the "indication" limitations of Asserted Claim 1 of the '885 Patent. *See* D.I. 309 at 6-11. My understanding is that Google is bound by this summary judgment ruling and precluded from challenging that an Accused Google Player programmed with the functionality represented by the foregoing source code path continues to operate in "standalone mode" after being added to a new speaker group and receiving "join_group" message indicating that the Accused Google Player has been added to the new speaker group.

92. Based on my analysis of the source code produced at SC-GOOG-SONOSNDCA-001598 - SC-GOOG-SONOSNDCA-001682, it appears to me that the functionality carried out by an Accused Google Player installed with newly-released firmware version 1.56.324896 when receiving a "join_group" message indicating that the Accused Google Player has been added to a new speaker group is very similar to the functionality that is represented by the foregoing source code path. In fact, the only relevant change I have observed in the source code is that MultizoneManager::RefreshDeviceGroups() function now calls an additional function named MultizoneManager::StopCurrentApp() (or "StopCurrentApp()" for short) prior to calling the MultizoneManager::AddGroup() function. This is illustrated by the following excerpt of the MultizoneManager::RefreshDeviceGroups() function that is reproduced in Dr. Schonfeld's

Rebuttal Report:



```
base::flat_set<std::string> group_uuids({virtual_group_uuid_});
  for (const auto& g : local_groups) {
    group_uuids.insert(g.uuid);
    auto it = groups_.find(g.uuid);
    if (it == groups_.end()) {
      StopCurrentApp();
      AddGroup(g);
    } else if (it->second->Reconfigure(g)) {
      SaveGroupConfig(g);
    } else {
      continue;
    }
    groups_changed_ = true;
```

Schonfeld Rebuttal Report at ¶ 59 (citing SC-GOOG-SONOSNDCA-001637 – 38); *see also* 1/25/2023 K. MacKay Rough Dep. Tr. at 9:19-11:4, 23:13-16 (Google's corporate designee testifying that the only change made to the RefreshDeviceGroups() function relative to prior versions of the source code was the addition of the StopCurrentApp() function and also confirming that there were no changes made to the AddGroup() function called by the RefreshDeviceGroups() function relative to prior versions of the source code); Sonos Dep. Ex. 1320-1321.

93. In this respect, my understanding of how this additional call to the MultizoneManager::StopCurrentApp() function impacts the functionality of an Accused Google Player installed with newly-released firmware version 1.56.324896 is that, if such an Accused Google Player is running a particular receiver app at the time that it receives "join_group" message indicating that the Accused Google Player has been added to a new speaker group (e.g., the YouTube Music receiver app), the MultizoneManager::StopCurrentApp() function will cause the Accused Google Player to stop that particular receiver app. In this respect, if the particular receiver app being run by the Accused Google Player is currently causing the Accused Google Player to engage in active playback, then the MultizoneManager::StopCurrentApp() function will cause the Accused Google Player to stop that active playback, whereas if the receiver app being run by the Accused Google Player is not currently causing the Accused Google Player to engage in active playback, then the MultizoneManager::StopCurrentApp() function will not impact the playback state of the Accused Google Player. However, in either case, the additional call to the MultizoneManager::StopCurrentApp() function does not cause an Accused Google Player

operating in a standalone mode to transition into a grouped mode in which it operates in accordance with the new speaker group. Rather, the MultizoneManager::StopCurrentApp() function causes an Accused Google Player operating in a standalone mode to stop its currently-running receiver app (to the extent that there is a receiver app currently running) while the new speaker group remains in an unlaunched state and the Accused Google Player remains in standalone mode.

94. My understanding of the MultizoneManager::StopCurrentApp() function has been confirmed by the testimony of Google's corporate designee, Mr. Kenneth MacKay. *See, e.g.*, 1/25/2023 K. MacKay Rough Dep. Tr. at 15:22-21:8 (testifying about the portion of the RefreshDeviceGroups() function where the StopCurrentApp() function is called), 23:17-33:9 (describing the operation of the StopCurrentApp() function), 33:10-13 (confirming that the StopCurrentApp() function will not "perform any checking of group state as part of stopping the app"), 33:14-34:1 (confirming that there is "no group information" that is passed into the StopCurrentApp() function), 34:2-9 (confirming that the StopCurrentApp() function does not cause any speaker group to be launched); 41:22-48:2 (describing the operation of the StopCurrentApp() function and how it differs from the StopApp() and StopPlayback() functions), 48:12-55:9, 56:16-62:14 (describing the operation of the StopCurrentApp() function in different scenarios for creating and modifying speaker groups).[7]

95. Moreover, the other functions in the source code path for receiving and handling a "join_group" message for a new speaker group do not appear to meaningfully differ from the functions included in the foregoing source code path that was already found to infringe the "continuing to operate in the standalone mode" limitation of Asserted Claim 1 of the '885 Patent.

[7] Mr. MacKay also testified that there may be scenarios where the StopCurrentApp() function will not stop a receiver app on an Accused Google Player installed with newly-released firmware version 1.56.324896, such as a scenario where the Accused Google Player is running a "non-visible app," as well as scenarios where a stopped current app is immediately replaced by another app that was "pending" and/or "preloaded" in the background at the time that the StopCurrentApp() function executes. *See* 1/25/2023 K. MacKay Rough Dep. Tr. at 27:7-28:1, 31:6-32:1, 43:15-45:6. These scenarios provide further support for my opinions, because even under Dr. Schonfeld's apparent theory that no longer running an app amounts to leaving "standalone mode," there will be scenarios where an Accused Google Player has at least one app running (or at least loaded) after executing the StopCurrentApp() function.

> is added to a group that is not playing back music will stop playback when added to that group.

Schonfeld Rebuttal Report at ¶ 47. However, this statement is flawed for several reasons.

98. First, Dr. Schonfeld's suggestion that "speakers added to the group *no longer continue their previous activity* and instead either play back music (if that is what the group was doing) or stop playback to match the group's state of stopped playback" is incomplete and inaccurate. As set forth above in Section IX.A, there are a number of scenarios where Accused Google Players installed with newly-released firmware version 1.56.324896 "continue their previous activity" after being added to a speaker group. For instance, in any scenario where an Accused Google Player installed with newly-released firmware version 1.56.324896 is operating in a standalone mode and is not engaging in active playback at the time when it is added to a new speaker group, that Accused Google Player will "continue [its] previous activity" after being added to the speaker group by continuing to operate in standalone mode and continuing not to engage in active playback. Likewise, in any scenario where an Accused Google Player installed with newly-released firmware version 1.56.324896 is operating in a standalone mode and is not engaging in active playback at the time when it is added to a pre-existing speaker group that is unlaunched, that Accused Google Player will "continue [its] previous activity" after being added to the pre-existing speaker group by continuing to operate in standalone mode and continuing not to engage in active playback.

99. Second, for similar reasons, Dr. Schonfeld's suggestion that "[s]peakers added to a speaker group do not continue with their previous playback or *non-playback state* when added to a group" is incomplete and inaccurate. Again, as set forth above in Section IX.A, there are a number of scenarios where Accused Google Players installed with newly-released firmware version 1.56.324896 "continue with their previous . . . *non-playback state*" after being added to a speaker group, including but not limited to the scenarios mentioned in the preceding paragraph.

100. Third, Dr. Schonfeld's suggestion that when an Accused Google Player installed with newly-released firmware version 1.56.324896 is added to a new speaker group, the Accused Google Player "stop[s] playback to match the group's state of stopped playback" is an inaccurate

and misleading characterization of the functionality for creating a new speaker group. In scenarios where an Accused Google Player is added to a new speaker group being created, the group begins in an uninvoked state (or an unlaunched state in Google's terms) – not a "state of stopped playback" as Dr. Schonfeld contends – and the Accused Google Player makes no reference to the "group's state" when handling the "join_group" message indicating that the Accused Google Player has been added to the new speaker group.

101. Indeed, an Accused Google Player installed with newly-released firmware version 1.56.324896 carries out the same functionality for handling the "join_group" message that was previously carried out by Accused Google Players installed with prior firmware versions, which undisputedly did not involve any "match[ing]" of the "group's state," along with one additional call to the "StopCurrentApp()" function discussed above. However, this "StopCurrentApp()" function does not make any reference to the "group's state," let alone causes the Accused Google Player to "match the group's state of stopped playback" as Dr. Schonfeld contends. Instead, the "StopCurrentApp()" function merely causes the Accused Google Player to stop its current receiver app, to the extent that such a receiver app is running. In this respect, if the receiver app being run by the Accused Google Player is currently causing the Accused Google Player to engage in active playback, then the "StopCurrentApp()" function will cause the Accused Google Player to stop that active playback, whereas if the receiver app being run by the Accused Google Player is not currently causing the Accused Google Player to engage in active playback, then the "StopCurrentApp()" function will not impact the playback state of the Accused Google Player – but in either case, the additional call to the "StopCurrentApp()" function does not cause an Accused Google Player to "match the group's state of stopped playback." And in a scenario where the Accused Google Player is not currently running a receiver app, the "StopCurrentApp()" function will have no impact at all on the state of the Accused Google Player. The foregoing operation of the "StopCurrentApp()" function is confirmed by the testimony of Mr. MacKay cited above.

102. Turning to Dr. Schonfeld's discussion of the functionality for modifying a pre-existing speaker group that is encoded within newly-released firmware version 1.56.324896, Dr. Schonfeld begins that discussion by making the following statement:

133. Second, this non-infringement theory is premised on Dr. Schonfeld's opinion that when an Accused Google Player installed with firmware version 1.56.324896 is added to a new speaker group via an Accused Google Controller, the Accused Google Player stops operating in "standalone mode" and "immediately begins operating as a member of the group, for example by playing music or not playing music, which varies based on the current operation of the group." *See* Schonfeld Rebuttal Report at ¶ 88. I disagree.

134. As a starting point, newly-released firmware version 1.56.324896 has no impact on the observable behavior of an Accused Google Player in a scenario where the Accused Google Player is configured for individual playback (rather than grouped playback) but is not currently engaged in active playback at the time that it is added to a new speaker group. Indeed, in such a scenario, an Accused Google Player installed with newly-released firmware version 1.56.324896 behaves in the same way that Accused Google Players installed with prior firmware versions behaved, namely, the Accused Google Player handles the join_group message received from the Accused Google Controller without leaving "standalone mode" or making any change to the Accused Google Player's inactive playback state. Thus, because an Accused Google Player installed with newly-released firmware version 1.56.324896 behaves in the same way that Accused Google Players installed with prior firmware versions behaved in at least one of the infringing scenarios that was presented to the Court as part of Sonos's summary judgment motion, it is my opinion that Accused Google Players installed with newly-released firmware version 1.56.324896 still infringe Asserted Claim 1 of the '885 Patent for the same reasons discussed in the Court's July 21, 2022 order. *See* 1/25/2023 K. MacKay Rough Dep. Tr. at 50:25-51:24 (Google's corporate designee confirming that in a scenario where an Accused Google Player being added to a new speaker group is not running a current receiver app, there will be no change to the "operational behavior" of the player which is no different than what happens in the prior firmware versions in this scenario).

135. Moreover, even focusing on a scenario where an Accused Google Player installed with firmware version 1.56.324896 is configured for individual playback and is currently engaged in active playback at the time that it is added to a new speaker group, I disagree with Dr.

Schonfeld's opinion that such an such an Accused Google Player "*begins operating as a member of the group*, for example by playing music or not playing music," which is contradicted by the evidence I have reviewed.

136. For example, the testing I observed demonstrated that an Accused Google Player installed with firmware version 1.56.324896 does not start operating in a grouped mode when it is added to a new speaker group. *See* Section IX.A. This is confirmed by the fact that (i) the new speaker group is not launched or otherwise activated at the time it is created and (ii) the only observable change in the behavior of the Accused Google Player being added to the new speaker group is that if the Accused Google Player was engaging in active playback prior to being added to the new speaker group, it stops that active playback. *Id.* Thus, there was nothing in the testing I observed even suggesting that an Accused Google Player installed with firmware version 1.56.324896 starts operating in a grouped mode when it is added to a new speaker group.

137. Consistent with the testing I observed, the source code for firmware version 1.56.324896 shows that when an Accused Google Player is added to a new speaker group, the only difference in operation relative to prior firmware versions is that the Accused Google Player is now programmed to execute a "StopCurrentApp()" function. And as already explained above, this "StopCurrentApp()" function does not cause the Accused Google Player to start operating in a grouped mode or to otherwise engage in any behavior in accordance with a speaker group. Instead, the "StopCurrentApp()" function merely causes the Accused Google Player to stop its currently-running receiver app (to the extent that there is a receiver app currently running at the Accused Google Player). In this respect, if the Accused Google Player is currently engaging in active playback, then the "StopCurrentApp()" function will cause the Accused Google Player to stop that active playback, whereas if the Accused Google Player is not currently engaging in active playback, then the "StopCurrentApp()" function will not impact the playback state of the Accused Google Player – but either way, the "StopCurrentApp()" function will not cause the Accused Google Player to leave "standalone mode" and "immediately begin[] operating as a member of the

238. As an initial matter, Dr. Schonfeld does not offer an opinion as to whether Google has committed acts of indirect infringement under my understanding of the law. To reiterate, independent Asserted Claim 1 of the '885 Patent is directed to a "first zone player" programmed with the capability to carry out the functions recited in elements 1.5-1.10. Similarly, Asserted Claim 1 of the '966 Patent is directed to a "computing device" that is programmed with the capability to carry out the functions recited in elements 1.4-1.11. And independent Asserted Claim 9 of the '966 Patent, is directed to a "computer readable medium" provisioned with program instructions that are executable to cause a computing device to perform the functions recited in elements 9.1-9.8. I understand that when claims are drawn to *capability*, such as each of these independent claims (as well as each of the dependent claims that ultimately depend from these independent claims), that direct infringement occurs as soon as an entity makes, uses, sells, offers for sale, or imports into the United States a device that is programed with the functional capability to engage in all of the elements recited in the claim, even if that entity never causes the device to actually engage in the functional steps recited by the claim. In this respect, I understand that (unlike for method claims) actual performance of the underlying functional elements is not required for direct infringement of claims drawn to capability.

239. Similarly, I understand that to induce infringement, an entity must cause, aid, or encourage direct infringement by another. In particular, I understand that a party induces infringement when (i) the party acted with the intent to encourage, aid, instruct, or otherwise cause another party to commit an act or acts that, if done by that other party, would constitute direct infringement of at least one claim of an asserted patent; (ii) the party at the time had knowledge or was willfully blind as to the existence of the asserted patent and the fact that the party's actions would lead the other in question to infringe the asserted patent directly; and (iii) the other party directly infringed at least one claim of the asserted patent. Thus, in this respect, I understand that for claims drawn to capability (like the asserted claims), the act of encouraging, aiding, instructing, or otherwise causing another to commit direct infringement occurs when the inducing party encourages, aids, instructs, or otherwise causes another to make, use, sell, offer for sale, or import into the United States a device that is programed with the functional capability to engage in all of

the elements recited in the claim (as this is a sufficient condition for direct infringement to occur), and that it is not necessary for the inducing party to encourage, aid, instruct, or otherwise cause another to engage in actual performance of the underlying functional steps recited by the claim.

240. Dr. Schonfeld does not offer any opinion as to whether Google has engaged in acts of indirect infringement of each of the asserted claims of the '885 and '966 Patents under my understanding of the law, nor does he dispute that there has been indirect infringement under my understanding of the law.[18]

241. Instead, Dr. Schonfeld sets forth a theory based on what appears to be an alternate, incorrect, understanding of the law of indirect infringement that would require Google's actions of encouraging, aiding, instructing, or otherwise causing another to commit direct infringement to be encouraging, aiding, instructing, or otherwise causing another to carry out the specific functionality recited by the individual elements of the claims (i.e., actual performance of the functional elements of the claim). As I explained, this is a flawed understanding of the law of indirect infringement and indirect infringement does not require evidence that Google, for instance, encouraged others to operate the Accused Google Controller and Accused Google Players by actually creating and saving overlapping speaker groups and invoking those speaker groups for synchronous playback.

242. Even if this were a requirement, however, Google's marketing material and instructions to customers do, in fact, encourage performance of the specific functional steps recited by the asserted claims. As one example, GOOG-SONOSWDTX-00007122 is a support page

[18] I further note that Dr. Schonfeld does not dispute several factual matters that underpin my opinion that indirect infringement has occurred. In particular, Dr. Schonfeld does not dispute the evidence that I pointed to that shows (i) Google's customers have purchased and then used the Accused Google Players, (ii) Google's customers have installed firmware updates on the Accused Google Players, (iii) Google's customers have downloaded and installed (or updated a previous installation of) the Google Home App onto computing devices that they purchased from Google or obtained from a third party, (iv) Google requires users (and instructs them) to download and install the Google Home App onto computing devices in order to set up, configure, and use the Accused Google Players, (v) Google has encouraged customers to download, install, and then use the Google Home App on computing devices, (vi) Google encourages customers to purchase and then use the Accused Google Players, and (vii) Google encourages or forces the Accused Google Players to receive firmware updates.

published by Google that encourages customers to "[g]roup any combination of Google Nest or Google Home speakers and displays and Chromecast devices together for synchronous music throughout the home." "[A]ny combination" would be understood to mean the types of combinations articulated by the claims, which include two overlapping speaker groups, where one Accused Google Player is a member of both of these speaker groups. The page goes on to explain and encourage exactly how to create a speaker group by articulating a series of five steps for operating the Google Home App to create a speaker group. GOOG-SONOSWDTX-00007122. As I explained in my opening report, these are the steps that establish that the Accused Google Controllers and the Accused Google Players are programmed with the functional capability to engage in the steps recited by the Asserted Claims. Accordingly, even under Dr. Schonfeld's erroneous understanding of the law, Google has engaged in acts of induced infringement.

2. <u>Contributory Infringement</u>

243. Dr. Schonfeld provides some additional (flawed) opinions under the header "Dr. Almeroth Has Not Established Contributory Infringement of the '885 Patent or '966 Patent" – although these opinions appear to concern indirect infringement generally as opposed to contributory infringement specifically. I disagree with his opinions and conclusions and provide explanations below where relevant. For all the reasons set forth in my opining report, my opinion remains that Google has and is engaging in acts of induced and contributory infringement.

244. Dr. Schonfeld opines that Google has not engaged in acts of contributory infringement for at least the reason that "knowledge of the asserted patents itself is not enough to meet the specific intent requirement." 1/13/2023 Schonfeld Rebuttal Report at ¶ 81. However, the evidence I identified in my 11/30/2022 Opening Report indicates that Google had both (i) knowledge of the Asserted Patents, and (ii) knowledge of how and why the Accused Google Players and the Google Home App practiced the asserted claims via at least Sonos's infringement contentions and allegations set forth in Sonos's pleadings documents and throughout the litigation. Additionally, the evidence I identified in my 11/30/2022 Opening Report indicates that Google engaged in the articulated acts of contributory (and induced) infringement with this specific knowledge, and with the further knowledge that its customers were engaging in acts that Google

knew to result in direct infringement of the asserted claims (e.g., making and using the Accused Google Controllers and the Accused Google Players). Thus, I disagree with Dr. Schonfeld's implication that Google lacked the specific intent required for indirect infringement.

245. Dr. Schonfeld opines that "I understand that Google had strong defenses and therefore good reason to believe it did not infringe the asserted patents." 1/13/2023 Schonfeld Rebuttal Report at ¶ 81. Dr. Schonfeld does not explain the basis for this understanding or articulate what the "good reason" was or is. Instead, he summarily cites to a number of disparate documents, such as Google's "declaratory judgment filing," "Answers to Sonos's Counterclaims," and its "interrogatory documents." Dr. Schonfeld fails to even purport to analyze these documents or cite to anything within these documents that would constitute any reasonable belief by Google that the Accused Google Controllers and Accused Google Players did not practice the asserted claims and thus that the articulated acts of indirect infringement by Google were not resulting in acts of direct infringement with Google's knowledge.

246. To the contrary, I am not aware of any reasonable belief by Google (Dr. Schonfeld has articulated none) that the Accused Google Controllers and the Accused Google Players did not practice the asserted claims. In fact, every purported non-infringement position that Google had concerning Asserted Claim 1 of the '885 Patent was rejected on summary judgment by the Court. In finding summary judgment of infringement, I understand that the Court concluded that no reasonable jury could have found non-infringement under Google's non-infringement positions. This establishes that these non-infringement positions were not "strong defenses" or "good reason" to believe Google did not infringe. And as I have outlined above, Google's positions concerning the Asserted Claims of the '966 Patent are similarly meritless. Further, after summary judgment of infringement was entered, Google did stop encouraging third parties to (i) offer to sell, sell, make, use, and/or import the Accused Google Players, which are the activities that constitute direct infringement of Asserted Claim 1 of the '885 Patent, or (ii) make and/or use the Accused Google Controllers, which are the activities that constitute direct infringement of the Asserted Claims of the '966 Patent.

247. Dr. Schonfeld then takes issue with my contention that the software components

supplied by Google (e.g., the software components of the Google Home App software package that Google provides to customers via app stores and the software components of the firmware updates Google provides to Accused Google Players) have no non-infringing uses. Dr. Schonfeld purports to address my no non-infringing use opinions, but points only to Section IX (which addresses the accused functionality of creating speaker groups) and Sections XIV and XVII (which address actual or hypothetical changes to the accused functionality) of his 1/13/2023 Rebuttal Report. None of these sections dispute my opinion that the accused software components have no non-infringing use because their only use is to be installed onto accused computing devices (e.g., a firmware update installed onto an Accused Google Player or the Google Home App software package installed onto a computing device, such as a Pixel 6 mobile phone). Indeed, once these software components are installed onto a computing device, an infringing instrumentality has been made, and thus direct infringement has occurred, as I have explained above and in my 11/30/2022 Opening Report. Based on my understanding of the governing legal standard, it is not relevant to the analysis whether the device, already installed with the accused software components, can engage in functionality that is not recited in the asserted claims (e.g., functionality related to Google Assistant, or functionality related to establishing a WiFi hotspot) because the device already infringes by virtue of having been installed with the accused software components, which, as soon as the installation takes place, renders the device with the functional capability to carry out the functions recited in the asserted claims and thus completes an act of direct infringement.

248. In sum, there is no non-infringing uses for the software components other than to be installed on computing devices, which is an act of direct infringement. Dr. Schonfeld does not dispute this or even address it.

XIV. <u>IMPORTANCE AND TECHNOLOGICAL BENEFITS OF THE CLAIMED TECHNOLOGY</u>

249. In my 11/30/2022 Opening Report, I set forth my opinion that both Google and its customers derive substantial benefit from Google's use of the claimed technology of the '885 and '966 Patents in the Accused Google Controllers and Accused Google Players. *See* 11/30/2022 Almeroth Opening Report at ¶¶ 706-730. Despite Google's own website, internal documents, and